# Creating Small Unit Based Glyph Visualisations

James R. Jackson*
Bangor University, UK

Panagiotis D. Ritsos†
Bangor University, UK

Jonathan C. Roberts‡
Bangor University, UK

## Abstract

Many modern day tasks involve the use of small screens, where users want to see a summary visualisation of an activity. For example, a runner using a smart watch needs to quickly view their progress, heart rate, comparison to previous races, etc. Subsequently, there is a need to portray data to users in small, yet well-defined, spaces. We define this space to be a single self-contained "unit". In this paper we introduce a glyph visualisation algorithm that creates a diverse range of visualisation designs; each design contains many separate parts, whereupon different parameters can be mapped. Our algorithm uses a path based approach which allows designers to create deterministic, yet unique designs, in a unit space to display multivariate data.

**Keywords:** Glyph, visualisation, mobile, small screen, wearable.

## 1 Introduction

In many settings users want to see a summary view of their information. Glyphs provide an ideal structure to summarise multivariate data. In addition, it is useful to have several unique glyph designs (or allow users to personalise the design for their purpose). We have been working with a healthcare provider who has requested that the summary information is displayed differently for several purposes. For example, a nurse may want to have a glyph design for self care, a different one for minimal care patients and another for severe patient care. The glyphs would be easily recognisable and visually distinguishable. In other situations, users may wish to create their own personalised glyph. Much like a logo is created specifically for a company, so a user would want to create their own glyph design. But designing a glyph is not an easy task for a user to achieve. Consequently, we have been investigating different ways to create glyphs and unit-based visualisations. In this short paper we introduce a path-based algorithm for generating deterministic glyph visualisations and present several results of the algorithm.

## 2 Background & Related Work

Glyphs have been used widely across data visualisation. For example, Ward [9] looked at glyph placement strategies, Roberts and Franklin [7] investigated haptic glyphs, and Borgo et al. [2] surveyed design guidelines, techniques and algorithms. A typical definition is that *Glyphs* are inter-connected shapes (or a single shape) that are used to depict a piece of pieces of related data [2]. We make a small extension to this idea. In our work we allow the parts of the glyph to include non visually connected elements. We view this idea in a similar way the public would observe a company Logo. A company logo would be easily recognisable as being separate to anything else on the page, yet it may contain separate non-overlapping parts. E.g., a logo may appear on a document, but the reader easily observes that it is different to the surrounding text. It is the whole design, appearance and colour combination that enables a human to view it as a single unit.

There is a growing need for the effective portrayal of data in visual form on mobile and small screen devices [4]. Patterson [6] posits that the primary objective of information design is to offer clear communication of data. As such we must strive to create visualisations which are well designed and produced and, ultimately, well interpreted. One of our motivating use-cases is that the glyph would be displayed in the full smart watch screen; glyphs with separate parts is suitable here, because the bevel of the watch visually encloses the glyph design. When our glyph is placed on alongside other information, such as in the nurse application, we would need to make sure that our glyph is visually separable from its background. Subsequently, we define our designs to be small unit based glyph visualisations. The portrayal of information within our "glyph units" can be achieved through a number of visual variables, including texture, size, shape, value, orientation and colour [1, 3].

## 3 Underpinning concepts and algorithm design

To research our algorithm we performed an in-depth study of glyph designs. We have recorded thousands of different designs and sketched many others using the Five Design-Sheet methodology [8]; we came to realise that we read the glyphs in an ordered way. We read circular glyphs starting at the top and going round clockwise; barchart glyphs we read left to right, etc. From this starting point, we started to explore how glyphs could be created from an ordered set of points. This set of order points we call a Path. Our idea of a "path" is similar to "path data" in the Scalable Vector Graphis definition. E.g., a triangle in SVG can be defined by $< path\ d =$ "$M$ 100 100 $L$ 300 100 $L$ 200 300 $z >$".

Figure 1 describes our method. We start by placing a series of equally spaced locations in the design space to designate a *path*. In the Figure we name the locations 1,2,3,4 and the edges A, B, C etc. Eventually we will map the data to these edges. We keep the order of the points contiguous, such that we can map them appropriately to their respective data values. Consequently, for a dataset with five data values we will produce six locations in the design space. Once these locations have been created we can start to adjust their positions and add other geometry to encode the data values.

The example in Figure 1 uses a dataset with the data points A, B and C. Each point pair represents a data point, with the next pair starting with the last point of the previous pair ($A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $\ldots$, $n-1 \rightarrow n$). We add the data values by adapting the type and size of geometry that we place along the edge. I.e., we adapt the envelope of the path. For instance, we could place a circle or a rectangle, or a complex polygon along each edge. By changing the geometry, constraining points of the geometry, or changing their appearance we can alter the final glyph design. For example, by keeping the path points in a straight line and adding rectangles along each edge (with the height of each rectangle mapped to data) we can make a bar chart. By placing the path points in a circle and using triangles with the centre point constrained to the centre of the unit we can mimic a star glyph. By randomly changing the position of the points and using rectangles we can mimic a Mondrian painting. Figure 1 shows four location points to define three parameter values. In this example we map the data to three ellipses, and change the colour of each ellipse for every parameter.

The designer can now change several variables to create different designs. These include the adjustment of the original points, envelope geometry, envelope colour, and constraints on different parts of the envelope geometry. As with placement the adjustment of the various properties of the glyph element are dictated by the glyph designer. Different properties can be changed such as width,

*e-mail: j.r.jackson@bangor.ac.uk
†e-mail: p.ritsos@bangor.ac.uk
‡e-mail: j.c.roberts@bangor.ac.uk

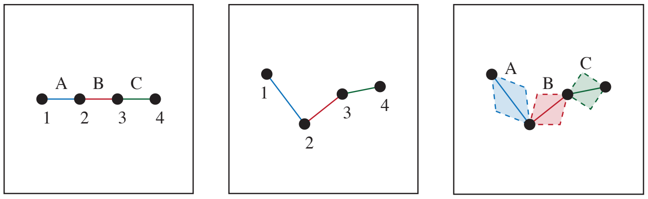


Fig. 1: (Left) The path is generated by placing equidistant (along a chosen axis) points within the design space. (Middle) Then the position of the points are adjusted making sure they keep within the 2d space. (Right) finally an envelope is added to the edges. Data values are mapped onto the envelope shape.

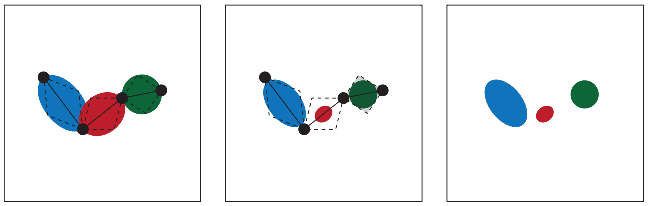

Fig. 2: In the example an elliptical element is placed with its centre point at the centre of the envelope. The size of the width (parallel to the line) and height (perpendicular to the line) of the elliptical is adjusted to depict the size of the datapoint

height, colour, opacity, texture, and orientation of the shapes. It is also useful to normalise the data values such that they appear suitably sized in the glyph unit. In Figure 2 we see that the height and width of the element are decreased to 80% of the available space.

We have explored several strategies to control envelope overlaps, including decreasing neighbouring envelope sizes, cutting out or obfuscating the intersection, showing only the intersection (that would be useful for continuous data), or by using the proximity to neighbouring elements, such to mimic a metaballs design [5].

## 4 Implementation and Results

We have implemented the algorithm in JavaScript using SVG. Figure 3 shows a showcase of different unit glyphs generated by our algorithm. The Figure demonstrates that we can create a range of different design ideas. We can create many more design ideas, but we are unable to showcase more because we are restricted by space in this short paper. Our pictures also demonstrate that our algorithm works within a unit square. We acknowledge that some mobile smart watches have circular screens, and we are looking to adapt the algorithm to work within a circular space. We have also used test data to map onto the designs. It is clear that some designs are better than others at encoding the data values, and additionally some look more "beautiful" than others. We are currently performing a large user study evaluating our glyph designs.

## 5 Discussion & Conclusions

This work is part of a large project investigating how glyphs can be displayed on mobile tablets and small screen devices, such as smart-watches. We believe that it is vital to have a consistent design strategy for glyph generation, and it is essential to allow glyphs to have non-conneted elements. Our path based design strategy outlined in this works allows developers to create glyphs quickly and easily while maintaining a rigid structure and pattern. Adoption of this method would allow users already familiar with path based languages (such as SVG path) to quickly and easily understand new visualisations. Our current implementation does have some limitations; we have not explored the rotation of objects along their path sections. Implementing this in future work would allow for a further visual channel with which to depict data. The generated exam-

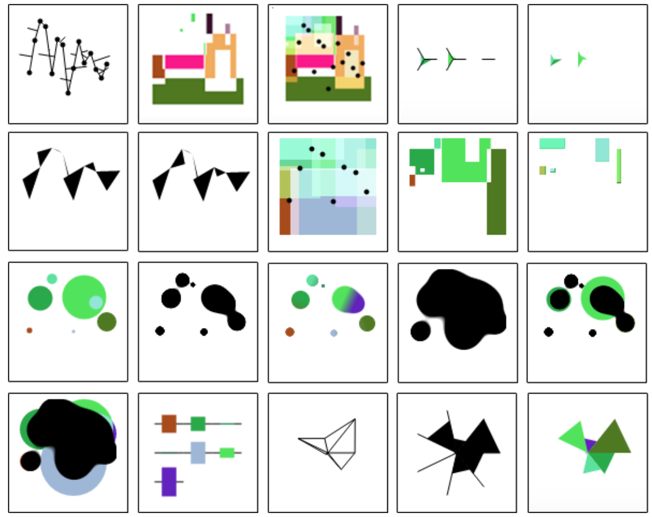

Fig. 3: This is a selection of the glyphs generated by the web based implementation. The top two rows depict the raw lines and envelopes

ples move path points at random distances on the y-axis, it would be possible to use the path to depict some continuous data within the dataset (such as a line graph). This could be achieved by making the envelope consider continuous data. Glyph elements could follow the design principles while following the envelope line. We have demonstrated that our path based method does allow for the creation of a vast array of unit glyph visualisations. The current glyphs are able to depict data in terms of size of object, colour and placement of line section, but we are unable to show this feature in detail, in this short paper due to space restrictions. Furthermore, we are exploring the effectiveness of the generated glyphs in terms of suitability, readability and desirability.

## 6 Acknowledgements

This work is funded by KESS 2. Knowledge Economy Skills Scholarships (KESS 2) is a pan-Wales higher level skills initiative led by Bangor University on behalf of the HE sector in Wales. It is part funded by the Welsh Governments European Social Fund (ESF) convergence programme for West Wales and the Valleys.

## References

[1] J. Bertin. *Graphics and graphic information processing*. Walter de Gruyter, 1981.

[2] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)*, pages 39–63, 2013.

[3] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *J. Amer. stat. assoc.*, 79(387):531–554, 1984.

[4] M. Lebusa, H. Thinyane, and I. Sieborger. Mobile visualisation techniques for large datasets. In *Proc. IST-Africa*, pages 1–9, May 2015.

[5] H. Nishimura. Object modeling by distribution function and a method of image generation. *Inst Electron Commun Eng Japan*, 68:718, 1985.

[6] R. Pettersson. Information design–principles and guidelines. *Journal of Visual Literacy*, 29(2):167–182, 2010.

[7] J. C. Roberts and K. Franklin. Haptic glyphs (hlyphs) - structured haptic objects for haptic visualization. In *Proc. World Haptics*, pages 369–374, March 2005.

[8] J. C. Roberts, C. Headleand, and P. D. Ritsos. Sketching designs using the five design-sheet methodology. *IEEE Trans Vis. Comp. Graph.*, 22(1):419–428, Jan 2016.

[9] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Info. Vis.*, 1(3/4):194–210, Dec. 2002.